

Hull 5-4-1-4

**RECEIVED
CENTRAL FAX CENTER**

JUN 29 2004

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**Patent Application**

Applicant(s): Hull et al.
Case: 5-4-1-4
Serial No.: 09/251,998
Filing Date: February 19, 1999
Group: 2143
Examiner: David E. England

I hereby certify that this paper is being deposited on this date with the U.S. Postal Service as first class mail addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

Signature: *[Signature]* Date: February 10, 2004

Title: Eager Evaluation of Tasks in a Workflow System

OFFICIAL**SUPPLEMENTAL APPEAL BRIEF**

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Supplemental Appeal Brief is submitted in response to the Office Action dated November 10, 2003 in the above-referenced application, in which the Examiner reopened prosecution in response to the Appeal Brief filed August 13, 2003.

Applicants have submitted concurrently herewith a response to the Office Action, requesting reinstatement of the appeal.

REAL PARTY IN INTEREST

The present application is assigned to Lucent Technologies Inc., as evidenced by an assignment recorded on May 21, 1999 in the United States Patent and Trademark Office at Reel 9970, Frame 0945. The assignee, Lucent Technologies Inc., is the real party in interest.

RELATED APPEALS AND INTERFERENCES

A Notice of Appeal was filed on October 21, 2003 in related United States Patent Application, Serial No. 10/274,579, and an Appeal Brief was filed in that application on December 23, 2003.

Hull 5-4-1-4

STATUS OF CLAIMS

The present application was filed on February 19, 1999, with claims 1-31. In a response to a restriction requirement, Applicants elected to prosecute claims 1-21. Consequently, claims 1-21 are currently pending. Claims 1 and 12 are independent claims. Claims 2-11 depend from independent claim 1, while claims 13-21 depend from independent claim 12.

Claims 1-21 stand rejected under the judicially created doctrine of obvious-type double patenting as being unpatentable over claims 1-28 of U.S. Patent No. 6,424,948. Claims 1-3, 5, 9, 12-14 and 16 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Hoenninger et al. (U.S. Patent No. 6,260,058, hereinafter "Hoenninger") in view of Rogers et al. (U.S. Patent No. 6,260,058, hereinafter, "Rogers"). Claims 4, 6, 7, 8, 15 and 17-19 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Hoenninger in view of Rogers and in further view of Boutaud et al. (U.S. Patent No. 6,253,307, hereinafter, "Boutaud"). Claims 10, 11, 20, and 21 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Hoenninger in view of Rogers and in further view of Van Praet et al. (U.S. Patent No. 5,854,929, hereinafter, "Van Praet") and Smith et al. (U.S. Patent No. 5,561,762, hereinafter, "Smith").

Claims 1-21 are appealed.

STATUS OF AMENDMENTS

There have been no amendments filed subsequent to the final rejection.

SUMMARY OF INVENTION

In a workflow system of the present invention, an object is processed through execution of a number of tasks. An exemplary workflow system is shown in FIG. 1 of the drawings and is described on page 15, line 16 to page 9, line 18 of the specification. This workflow system is an object-focused workflow system that processes objects, which may be organized as modules. See, for example, page 6, lines 14-21. Modules have a number of enabling conditions associated with them. The enabling conditions indicate whether a module is to be executed for the object. FIG. 2

Hull 5-4-1-4

shows a ROUTING_TO_SKILL module having a number of other modules with associated enabling conditions. FIG. 2 is described on page 9, line 19 to page 13, line 8.

Tasks are associated with modules and are referred to by their associated modules. Tasks are described, e.g., on page 39, lines 11-17. Execution of one or more of the tasks results in initiation of a side-effect action performed by a component external to the workflow system. Side-effect actions are described, for instance, in reference to FIG. 4 and on page 13, line 21 to page 14, line 4. It is determined whether a task is eligible for eager execution by considering at least (1) a state of the task and (2) whether execution of the task results in the initiation of a side-effect action. The task is executed using eager execution if the task is determined to be eligible for eager execution. States of tasks are described, for instance, at page 40, line 20 to page 41, line 8 and page 64, lines 11-22. Algorithms for determining states of tasks and whether tasks should be executed eagerly are described in FIGS. 34A-34D and 35A-35D and associated text (e.g., page 41, line 9 to page 64, line 22).

ISSUES PRESENTED FOR REVIEW

1. Whether claims 1-21 are properly rejected under the judicially created doctrine of obvious-type double patenting as being unpatentable over claims 1-28 of U.S. Patent No. 6,424,948;
2. Whether claims 1-3, 5, 9, 12-14 and 16 are properly rejected under 35 U.S.C. §103(a) as being unpatentable over Hoenninger in view of Rogers;
3. Whether claims 4, 6, 7, 8, 15 and 17-19 are properly rejected under 35 U.S.C. §103(a) as being unpatentable over Hoenninger in view of Rogers and in further view of Boutaud; and
4. Whether claims 10, 11, 20, and 21 are properly rejected under 35 U.S.C. §103(a) as being unpatentable over Hoenninger in view of Rogers and in further view of Van Praet and Smith.

GROUPING OF CLAIMS

The rejected claims do not stand or fall together.

With regard to Issue (1), claims 1-21 stand or fall together.

Hull 5-4-1-4

With regard to Issue (2), claims 1, 9 and 12 stand or fall together, claims 2 and 13 stand or fall together, claims 3 and 14 stand or fall together, and claims 5 and 16 stand or fall together.

With regard to Issue (3), claims 4 and 15 stand or fall together, claims 6 and 17 stand or fall together, claims 7 and 18 stand or fall together, and claims 8 and 19 stand or fall together.

With regard to Issue (4), claims 10, 11, 20 and 21 stand or fall together.

ARGUMENT

Issue (1)

As to Issue (1) presented above, the Examiner rejected claims 1-21 under the judicially created doctrine of obvious-type double patenting as being unpatentable over claims 1-28 of U.S. Patent No. 6,424,948. Applicants respectfully traverse this rejection.

Independent claims of U.S. Patent No. 6,424,948 have limitations of executing a program specification, wherein the step of executing further comprises the steps of evaluating enabling conditions and executing modules based on said evaluation of enabling conditions. Independent claims 1 and 12 of the present invention contain limitations of determining whether a task is eligible for eager execution by considering at least (1) a state of the task and (2) whether execution of the task results in the initiation of a side-effect action, and executing the task using eager execution if the task is determined to be eligible for eager execution.

Although "side-effect actions" are referred to in certain claims of U.S. Patent No. 6,424,948, there is no mention of "eager execution" of tasks in the claims of U.S. Patent No. 6,424,948, of the "state" of a task or of "determining whether a task is eligible for eager execution by considering at least (1) a state of the task and (2) whether execution of the task results in the initiation of a side-effect action," as found in independent claims 1 and 12 of the present invention. Therefore, the limitations of independent claims 1 and 12 of the present invention are not obvious in light of the claims of U.S. Patent No. 6,424,948.

Hull 5-4-1-4

For the reasons given above, Applicants respectfully submit that the rejection of claims 1-21 under the judicially created doctrine of obvious-type double patenting, as being unpatentable over claims 1-28 of U.S. Patent No. 6,424,948, is improper.

Issue (2)

The Examiner rejected claims 1-3, 5, 9, 12-14 and 16 under 35 U.S.C. §103(a) as being unpatentable over Hoenninger in view of Rogers.

Regarding independent claims 1 and 12, the Examiner asserts that Hoenninger does not teach determining whether execution of a task results in the initiation of a side-effect action, and executing the task using eager execution if the task is determined to be eligible for eager execution, but that Rogers teaches these limitations. In particular, the Examiner cites col. 11, line 45 to col. 12, line 16 and col. 15, lines 44-52 of Rogers in support of the assertion that Rogers teaches the limitation of whether execution of a task results in the initiation of a side-effect action. Further, the Examiner points to col. 29, lines 1-28 and col. 37, line 5 to col. 38, line 52 of Rogers for the assertion that Rogers teaches the limitation of executing the task using eager execution if the task is determined to be eligible for eager execution.

Applicants respectfully submit that Rogers does not teach the limitations of "determining whether a task is eligible for eager execution by considering . . . whether execution of the task results in the initiation of a side-effect action" and "executing the task using eager execution if the task is determined to be eligible for eager execution." Applicants note that both independent claims 1 and 12 contain the limitation of "execution of at least one of said tasks results in initiation of a side-effect action performed by a component external to said workflow system" and that a "side-effect action" is further defined in the present specification at page 13, line 21 to page 14, line 24.

Applicants read Rogers as providing a call management system for management of calls by system users at workstation computers. See, for instance, Abstract of Rogers. The Examiner appears to assert that a call type in Rogers is a type of task. See paragraph 13 of the outstanding Office Action (e.g., "call types as types of

Hull 5-4-1-4

tasks"). Applicants respectfully submit that even if a "call type" in Rogers is a type of task, Rogers does not determine whether a task is eligible for eager execution by considering whether execution of the task results in the initiation of a side-effect action.

Rogers does determine a "call type" of an incoming call type of a call, such as voice, fax or data. See col. 11, lines 45-50. The determination of call type is performed in parallel (see FIG. 5 of Rogers). If the call is fax or data, the call is handled by the appropriate protocol (see FIG. 5, step 503 of Rogers). If the call is a voice call to an identified party, then "VIP rules" are applied (see FIG. 5, steps 502 through 521 of Rogers). VIP rules are call handling rules defined by a business organization or its users. See col. 2, lines 22-27 of Rogers. Additionally, if a call is a voice call type, then Rogers can create a voice pathway to an in-house or external computer user. See col. 15, lines 44-52 and col. 2, lines 50-55 of Rogers.

For sake of argument, if a "call type" in Rogers is defined as a "task" in accordance with independent claims 1 and 12 of the present invention, then Rogers does not determine whether the call type (i.e., "task") is eligible for eager execution by considering whether execution of the call type (i.e., "task") results in the initiation of a side-effect action, because a call type is simply a type of call in Rogers and is not "executed." See, for instance, FIG. 5 of Rogers.

As another example for sake of argument, if a "voice pathway" in Rogers is defined as a "task" in accordance with independent claims 1 and 12 of the present invention, then Rogers does not determine whether the voice pathway (i.e., "task") is eligible for eager execution by considering whether execution of the voice pathway (i.e., "task") results in the initiation of a side-effect action, because the voice pathway is simply created (or reused) in Rogers when a call is to be put through to a destination. See, for instance, col. 2, lines 50-55 of Rogers. Even if a voice pathway is "executed," Applicants can find no disclosure in Rogers that consideration is given to whether "execution" of a voice pathway (i.e., "task") results in initiation of a side-effect action as claimed in independent claims 1 and 12 of the present invention.

As yet another example, if a "VIP rule" in Rogers is defined as a "task" in accordance with independent claims 1 and 12 of the present invention, then Rogers does not determine whether the VIP rule (i.e., "task") is eligible for eager execution by

Hull 5-4-1-4

considering whether execution of the VIP rule (i.e., "task") results in the initiation of a side-effect action, because a VIP rule is simply applied in Rogers when a call is identified as originating from a VIP caller. See, for instance, col. 3, lines 52-65 of Rogers. Applicants respectfully submit that there is no disclosure in Rogers of considering whether "execution" of a VIP rule (i.e., "task") results in the initiation of a side-effect action as claimed in independent claims 1 and 12 of the present invention.

Because the Examiner admits that Hoenninger does not teach determining whether execution of a task results in the initiation of a side-effect action, and executing the task using eager execution if the task is determined to be eligible for eager execution, and because Applicants have shown that Rogers does not teach these limitations, then the combination of Hoenninger and Rogers cannot teach these limitations. Therefore, Applicants respectfully submit that independent claims 1 and 12 are patentable over Hoenninger and Rogers, alone or in combination. Because claim 1 is patentable, its dependent claim 9 is also patentable.

With regard to dependent claims 2, 3, 5, 9, 13, 14 and 16, the Examiner also rejected these claims under 35 U.S.C. §103(a) as being unpatentable over Hoenninger in view of Rogers.

With regard to claims 2 and 13, which stand or fall together, these claims contain the limitations of "determining that a particular task whose execution results in the initiation of a side-effect action is eligible for eager execution only if it is determined that the one or more enabling conditions associated with the particular task will evaluate to true as determined by the state of the particular task." The Examiner asserts that Hoenninger teaches this limitation and cites col. 6, line 33 to col. 7, line 42 of Hoenninger.

Applicants respectfully submit that, while Hoenninger does describe a number of states for a task in the cited text and a state diagram (see FIG. 3 of Hoenninger), there is no teaching in Hoenninger of determining that one or more enabling conditions associated with the particular task will evaluate to true as determined by the state of the particular task. Applicants define enabling conditions in independent claims 1 and 12 (from which dependent claims 2 and 13, respectively, depend) as "one or more of said tasks each having one or more associated enabling conditions indicating

Hull 5-4-1-4

whether the task is to be executed for said object." See also page 2, lines 24-26 of the present specification. In Hoenninger, a task can be assigned a "blocked" state when computation results from another task are not available (see col. 6, lines 53-55 of Hoenninger) or memory is not available for a computation (see col. 6, line 65 to col. 7, line 5 of Hoenninger), but there is no determination that an enabling condition for a task in Hoenninger will evaluate to true as determined by the state of the task. Even if Hoenninger does determine that an enabling condition for a task will evaluate to true, this determination is not made by using state of the task in Hoenninger.

Furthermore, Applicants respectfully submit that Rogers does not disclose enabling conditions that indicate whether a task is to be executed for an object and determining that a particular task whose execution results in the initiation of a side-effect action is eligible for eager execution only if it is determined that the one or more enabling conditions associated with the particular task will evaluate to true as determined by the state of the particular task, in accordance with independent claims 1 and 12 and dependent claims 2 and 13.

There is no disclosure in Hoenninger or Rogers of tasks that have enabling conditions as defined by and used in independent claims 1 and 12 and dependent claims 2 and 13. Consequently, Applicants respectfully submit that dependent claims 2 and 13 are patentable over Hoenninger or Rogers, alone or in combination.

With regard to claims 3 and 14, which stand or fall together, each of these claims has the limitations of "determining that a particular task whose execution does not result in the initiation of a side-effect action is eligible for eager execution prior to determining that the one or more enabling conditions associated with the particular task will evaluate to true as determined by the state of the particular task." The Examiner points to col. 29, lines 1-51, and col. 38, line 63 to col. 39, line 3 of Rogers for the disclosure of dependent claims 3 and 14.

Applicants respectfully submit that while the cited text of Rogers describes user call status and how actions are determined for a call, Applicants can find no disclosure in Rogers that Rogers determines that a task is eligible for eager execution prior to determining that an associated enabling condition will evaluate to true, as claimed in dependent claims 3 and 14. User call status in Rogers is simply the status of a

Hull 5-4-1-4

call (e.g., on hold, ringing, and connected), while the actions performed are actions such as receiving a call or transferring a call. Consequently, Applicants respectfully submit that dependent claims 3 and 14 are patentable over Hoenninger and Rogers, alone or in combination.

With regard to claims 5 and 16, which stand or fall together, these claims have the additional limitations of "wherein said step of determining whether a task is eligible for eager execution is performed by also considering (3) whether the task contributes to the production of a target value." The Examiner cites col. 2, lines 10-34 and col. 3, lines 9-65 of Rogers as disclosing these limitations.

Applicants respectfully disagree. The cited text in Rogers describes, for instance, VIP rules and their processing. For the sake of argument, if a "task" is considered to be a "VIP rule" in Rogers, there is no disclosure in Rogers describing that a determination of whether a VIP rule (i.e., "task") is eligible for eager execution by considering whether the VIP rule (i.e., "task") contributes to a target value, as VIP rules in Rogers are simply applied when appropriate. See, for instance, col. 2, lines 27-34 of Rogers. Consequently, Applicants respectfully submit that dependent claims 5 and 16 are patentable over Hoenninger and Rogers, alone or in combination.

Issue (3)

With regard to Issue (3), the Examiner rejected claims 4, 6, 7, 8, 15 and 17-19 under 35 U.S.C. §103(a) as being unpatentable over Hoenninger in view of Rogers and in further view of Boutaud.

With regard to claims 4 and 15, which stand or fall together, these claims add the limitation of "partially evaluating one or more enabling conditions associated with said task." The Examiner points to col. 45, line 58 to col. 46, line 51 of Boutaud as teaching this limitation. Applicants read the cited sections of Boutaud as describing "conditional instructions" that can be or not be executed based on a condition. See, for instance, col. 46, lines 13-25 of Boutaud. Applicants respectfully submit that the cited text of Boutaud does not disclose any item that is partially evaluated. For example, if a condition in Boutaud is true, then certain conditional instructions are executed; if a

Hull 5-4-1-4

condition in Boutaud is false, then those certain conditional instructions are not executed. See col. 46, lines 20-25 of Boutaud.

By contrast, an enabling condition of the present invention can be partially evaluated. For example, FIG. 20 shows the enabling condition "cust_value < 7 and DNIS not = 'Australia_promotion.'" The "cust_value < 7" part of the enabling condition could be evaluated, which means that only part of the enabling condition "cust_value < 7 and DNIS not = 'Australia_promotion'" would be evaluated. In Boutaud, the conditional instructions are either executed or not executed, and there is no partial evaluation of the conditional instructions.

Thus, Applicants respectfully submit that independent claims 4 and 15 are patentable over Hoenninger, Rogers and Boutaud, alone or in combination.

With regard to claims 6 and 17, which stand or fall together, these claims have the limitation of "determining that a particular task is unneeded for processing of the object based at least in part on partial evaluation of an enabling condition of a second task, wherein said second task's enabling condition depends on one or more outputs of said particular task." As described above, Applicants submit that Boutaud does not disclose a partial evaluation of an enabling condition. Moreover, in Boutaud if a "conditional instruction" is considered to be a "task," then there is no determination that a particular conditional instruction (i.e., "task") is necessary based on evaluation of an enabling condition of a second conditional instruction (i.e., "task"). Instead, in Boutaud, if a condition is or is not true, the "conditional instructions" are or are not executed, respectively, and conditional instructions do not depend on enabling conditions of other conditional instructions. Consequently, Applicants respectfully submit that dependent claims 6 and 17 are patentable over Hoenninger, Rogers and Boutaud.

Regarding claims 7 and 18, which stand or fall together, the same reasoning applies as in regards to dependent claims 6 and 17. Furthermore, if a "conditional instruction" in Boutaud is considered to be a "task," Boutaud does not disclose that a particular conditional instruction (e.g., "task") is based on evaluation of enabling conditions for a number of tasks, as conditional instructions in Boutaud either are or are not executed depending on a single condition. Consequently, Applicants

Hull 5-4-1-4

respectfully submit that dependent claims 7 and 18 are patentable over Hoenninger, Rogers and Boutaud.

Regarding claims 8 and 19, which stand or fall together, these claims add the limitations of "determining that a particular task is necessary for processing of the object based at least in part on evaluation of enabling conditions for a number of tasks, wherein said tasks' enabling conditions depend on one or more outputs of said particular task." Applicants submit that these claims are patentable for at least the reasons stated above with regard to claims 5, 6, 16 and 17. Furthermore, if a "conditional instruction" in Boutaud is considered to be a "task," then Boutaud does not disclose that a number of conditional instructions' (i.e., "tasks") enabling conditions depend on outputs of a particular conditional instruction (i.e., "task"), as conditional instructions in Boutaud either are or are not executed depending on a single condition. Consequently, Applicants respectfully submit that dependent claims 8 and 19 are patentable over Hoenninger, Rogers and Boutaud.

Issue (4)

With regard to Issue (4), the Examiner rejected claims 10, 11, 20, and 21 under 35 U.S.C. §103(a) as being unpatentable over Hoenninger in view of Rogers and in further view of Van Praet and Smith.

With regard to claims 10, 11, 20 and 21, which stand or fall together, each of claims 10 and 20 has the limitations of "a graph representing data flow dependencies and enabling flow dependencies between tasks and enabling conditions" and "propagating changes through said graph based on new outputs of completed tasks." Claim 11 depends from claim 10 and claim 21 depends from claim 20. Van Praet discloses a "bipartite" graph where vertices represent storage elements in a processor or operations of a processor, and where edges represent connectivity of a processor and data flow from storage. See col. 8, lines 51-57 of Van Praet. Smith discloses a graph where each node represents a logic gate and the branches represent input or output lines. See col. 5, lines 51-58 of Smith.

In the present invention, as described above, a task has one or more associated enabling conditions indicating whether the task is to be executed for an object

Hull 5-4-1-4

(see, e.g., independent claims 1 and 12). Furthermore, a task can produce an output that is used in an enabling condition for another task. See, for instance, FIG. 26 and associated text on pages 36 and 37 of the present specification, where it states the following:

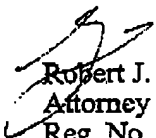
This diagram illustrates the data flow dependencies and the enabling flow dependencies of the workflow described above. Each of the modules (ovals) and enabling conditions (hexagons) are represented as nodes with solid line data flow edges representing data flow dependencies and broken line enabling flow edges representing enabling flow dependencies.

If a "task" of the present invention is a store element or processor of Van Praet, while Van Praet might, for sake of argument, show a data flow dependency in a graph, there is no disclosure of an enabling flow dependency in the graph. Similarly, if a "task" of the present invention is a logic gate, while Smith might, for sake of argument, show a data flow dependency in a graph, there is no disclosure of an enabling flow dependency in the graph. In other words, in both Van Praet and Smith, only one data dependency (e.g., "edge" or "connection") is shown between nodes, while claims 10 and 20 require two types of data dependencies. Consequently, Applicants respectfully submit that dependent claims 10 and 20 are patentable over Hoenninger, Rogers, Van Praet and Smith, alone or in combination. Because claims 10 and 20 are patentable, their respective dependent claims 11 and 21 are patentable.

Applicants respectfully submit that claims 1-21 of the present invention are patentable. The attention of the Examiner and the Appeal Board to this matter is appreciated.

Respectfully,

Date: February 10, 2004


Robert J. Mauri
Attorney for Applicant(s)
Reg. No. 41,180
Ryan, Mason & Lewis, LLP
1300 Post Road, Suite 205
Fairfield, CT 06430
(203) 255-6560

Hull 5-4-1-4

APPENDIX

1. A method for operation of a workflow system for processing an object by executing a plurality of tasks, one or more of said tasks each having one or more associated enabling conditions indicating whether the task is to be executed for said object, and wherein execution of at least one of said tasks results in initiation of a side-effect action performed by a component external to said workflow system, said method comprising the steps of:

determining whether a task is eligible for eager execution by considering at least (1) a state of the task and (2) whether execution of the task results in the initiation of a side-effect action; and

executing the task using eager execution if the task is determined to be eligible for eager execution.

2. The method of claim 1 wherein the step of determining whether a task is eligible for eager execution further comprises the step of:

determining that a particular task whose execution results in the initiation of a side-effect action is eligible for eager execution only if it is determined that the one or more enabling conditions associated with the particular task will evaluate to true as determined by the state of the particular task.

3. The method of claim 1 wherein the step of determining whether a task is eligible for eager execution further comprises the step of:

determining that a particular task whose execution does not result in the initiation of a side-effect action is eligible for eager execution prior to determining that the one or more enabling conditions associated with the particular task will evaluate to true as determined by the state of the particular task.

4. The method of claim 1 wherein said step of determining whether a task is eligible for eager execution further comprises the step of:

partially evaluating one or more enabling conditions associated with said task.

Hull 5-4-1-4

5. The method of claim 1 wherein said step of determining whether a task is eligible for eager execution is performed by also considering (3) whether the task contributes to the production of a target value.

6. The method of claim 1 further comprising the step of:
determining that a particular task is unneeded for processing of the object based at least in part on partial evaluation of an enabling condition of a second task, wherein said second task's enabling condition depends on one or more outputs of said particular task.

7. The method of claim 1 further comprising the step of:
determining that a particular task is necessary for processing of the object based at least in part on evaluation of enabling conditions for a number of tasks, wherein said tasks' enabling conditions depend on said particular task.

8. The method of claim 1 further comprising the step of:
determining that a particular task is necessary for processing of the object based at least in part on evaluation of enabling conditions for a number of tasks, wherein said tasks' enabling conditions depend on one or more outputs of said particular task.

9. The method of claim 1 wherein said step of determining is performed repeatedly during the processing of the object.

10. The method of claim 1 wherein a memory of said workflow system stores a graph representing data flow dependencies and enabling flow dependencies between tasks and enabling conditions, said method further comprising the step of:
propagating changes through said graph based on new outputs of completed tasks.

Hull 5-4-1-4

11. The method of claim 10 wherein said step of propagating changes is based on predefined propagation rules.

12. A workflow system for processing an object by executing a plurality of tasks, one or more of said tasks each having one or more associated enabling conditions indicating whether the task is to be executed for said the object, and wherein execution of at least one of said tasks results in initiation of a side-effect action performed by a component external to said workflow system, said system comprising:

means for determining whether a task is eligible for eager execution by considering at least (1) a state of the task and (2) whether execution of the task results in the initiation of a side-effect action; and

means for executing the task using eager execution if the task is determined to be eligible for eager execution.

13. The workflow system of claim 12 wherein the means for determining whether a task is eligible for eager execution further comprises:

means for determining that a particular task whose execution results in the initiation of a side-effect action is eligible for eager execution only if it is determined that the one or more enabling conditions associated with the particular task will evaluate to true as determined by the state of the particular task.

14. The workflow system of claim 12 wherein the means for determining whether a task is eligible for eager execution further comprises:

means for determining that a particular task whose execution does not result in the initiation of a side-effect action is eligible for eager execution prior to determining that one or more enabling conditions associated with the particular task will evaluate to true as determined by the state of the particular task.

15. The workflow system of claim 12 wherein said means for determining whether a task is eligible for eager execution further comprises:

Hull 5-4-1-4

means for partially evaluating one or more enabling conditions associated with said task.

16. The workflow system of claim 12 wherein said means for determining whether a task is eligible for eager execution further comprises:

means for determining whether the task contributes to the production of a target value.

17. The workflow system of claim 12 further comprising:

means for determining that a particular task is unneeded for processing of the object based at least in part on partial evaluation of an enabling condition of a second task, wherein said second task's enabling condition depends on one or more outputs of said particular task.

18. The workflow system of claim 12 further comprising:

means for determining that a particular task is necessary for processing of the object based at least in part on evaluation of enabling conditions for a number of tasks, wherein said tasks' enabling conditions depend on said particular task.

19. The workflow system of claim 12 further comprising:

means for determining that a particular task is necessary for processing of the object based at least in part on evaluation of enabling conditions for a number of tasks, wherein said tasks' enabling conditions depend on one or more outputs of said particular task.

20. The workflow system of claim 12 further comprising:

a memory for storing a graph representing data flow dependencies and enabling flow dependencies between tasks and enabling conditions; and

means for propagating changes through said graph based on new outputs of completed tasks.

Hull 5-4-1-4

21. The workflow system of claim 20 wherein said memory stores predefined propagation rules and wherein said means for propagating changes further comprises means for propagating changes based on said predefined propagation rules.